พอยน์เตอร์ในภาษาปาสคาล : สนุกกับจอภาพ (จบ)

บุญเลิศ เอี่ยมทัศนา

ในตอนที่แล้ว (ไมโครฯ 41) ผมได้กล่าวถึงการใช้งานพอยน์เตอร์ในการสร้างหน้าต่างแบบธรรมดา ในฉบับนี้จะเดินหน้าต่อไปครับ คือให้ผู้ใช้เลื่อนหน้าต่างไปมาได้ กับจะให้ผู้ใช้เลือกหน้าต่างที่ซ้อนกันอยู่ว่าจะให้หน้าต่างใดอยู่บน

เตรียมการ

ความจริงโครงงาน 3 ตอนนี้ คือโครงงานเดียวกัน แต่ผมต้องการแยกแยะในแต่ละเรื่อง จึงได้แยกออกจากกัน ดังนั้นจึงมีโปรแกรมอีกบางส่วนที่ใช้ร่วมกัน จึงขอให้ก็อบปี้โพรซีเดอร์ OpenWnd1to, CloseWindow และ CreateUserWindow ซึ่งอยู่ในรูป B และ C ของตอนที่แล้ว ไว้ในไฟล์ WINDOW2.ICL (ใช้ mark block กับ Ctrl-K,W)

เลื่อนหน้าต่าง

การที่ให้ผู้ใช้เลื่อนหน้าต่างได้ ก็เพื่อให้ผู้ใช้เลื่อนหน้าต่างบนไปทางด้านใดด้านหนึ่งเพื่อที่จะอ่านข้อความในหน้าต่างล่างที่บังกันอยู่ การเลื่อนหน้าต่างนี้อาจทำได้โดย ให้ผู้ใช้บอกจุดโคออร์ดิเนตของมุมบนซ้าย หรือให้กดคีย์ลูกศร คิดว่าการใช้คีย์ลูกศรจะสะดวกแก่ผู้ใช้มากกว่า

ขอให้ดูในรูป A2 ซึ่งก็คือรูป A ในฉบับที่แล้ว เป็นการกำหนดแบบและค่าคงตัวต่าง ๆ โดยเปลี่ยนชื่อโปรแกรม และได้ผนวก (include) ไฟล์ WINDOW2.ICL

การทำงานของโปรแกรมนี้ ขอเริ่มที่โปรแกรมหลักในรูป M2 เริ่มด้วยการเตรียมหน้าต่าง ตามค่า MAXWNDW

A2

```
PROGRAM  DemoPointer2(input,output);
USES     crt;

(*-----------------  CONSTANT  for WINDOW.ICL  -----------*)
CONST    MAXWNDW = 4;  MAXSTCK = 10;
TYPE     edgeT   = ARRAY[1..4] OF Byte;
         WindowSizeT = ARRAY[0..MAXWNDW] OF edgeT;
CONST    wndSize : WindowSizeT =
          ((1,80,1,25), (1,80,1,3), (1,80,10,24),
           (50,70,1,15), (1,80,1,25));
{$I WINDOW.ICL}
(*-------------------------------------------------------*)
TYPE     promptT  = ARRAY[1..MAXWNDW] OF String[13];
CONST    promp    : promptT =
          ('FIRST WINDOW ','SECOND WINDOW','THIRD WINDOW','');
{$I WINDOW2.ICL}
```

คือ 4 หน้าต่างกับของจอภาพ ต่อไปสมมุติว่าผู้ใช้ใช้งานหน้าต่าง 0 คือจอภาพอยู่ โดยให้ป้อนข้อความใด ๆ ก็ได้ไปจนกว่าผู้ใช้จะกดคีย์ ENTER ที่ต้นบรรทัด

ลำดับต่อไปเป็นการเปิดหน้าต่างขึ้น 3 หน้าต่างด้วย OpenWndIto(3) โดยจะเปิดหน้าต่างให้ผู้ใช้ใช้งานทีละหน้าต่าง (ป้อนข้อความจบด้วยการกดคีย์ ENTER ที่ต้นบรรทัดเช่นกัน) เมื่อจบหน้าต่างหนึ่งก็จะเปิดหน้าต่างต่อไป (การทำงานนี้ให้ดูได้ในโพรซีเดอร์ OpenWndIto รูป B ในไมโครฯ ฉบับที่ 41)

เสร็จแล้วจะให้ผู้ใช้กำหนดขนาดของหน้าต่างด้วยตนเองแล้วเปิดใช้ด้วยโพรซีเดอร์ CreateUserWindow (รูป C ฉบับที่แล้ว) ซึ่งผู้ใช้จะต้องใช้งานจนกดคีย์ ENTER ที่ต้นบรรทัดอีกเช่นกัน

เท่าที่กล่าวมานี้คือการใช้หน้าต่างตามที่กล่าวในฉบับที่แล้ว จากนี้ก็จะเป็นการเลื่อนหน้าต่างบน ซึ่งหน้าต่างบนในขณะนี้คือหน้าต่างที่ผู้ใช้เปิดไว้

แต่ผมต้องการชี้แจงให้ผู้ใช้ทราบว่าให้กดคีย์ลูกศรในการเลื่อนหน้าต่าง จึงเลือกเปิดหน้าต่าง 1 (จะเป็นหน้าต่างใดก็ได้) ซ้อนบนอีกชั้นด้วย OnTopWindow ให้เอาต์พุตข้อความที่ต้องการ แล้วปิดด้วย OffTopWindow ผลการรันโปรแกรมถึงขณะนี้ตามรูปที่ 1

ลำดับต่อไปให้ผู้ใช้เลื่อนหน้าต่างนี้ได้ด้วยคีย์ลูกศรทั้งสี่ ด้วย MoveWindowAround ซึ่งผมได้เลื่อนไปไว้ทางล่างขวาของเดิมตามรูปที่ 2 เมื่อพอใจแล้วให้กดคีย์ Enter โปรแกรมนี้ไม่ได้ดำเนินการใดต่อ จึงให้ปิดหน้าต่างทั้งสี่ด้วย CloseNwindow สำหรับการทำงานของโปรแกรมย่อยที่สร้างใหม่มีดังนี้

OnTopWindow (n)

เป็นการเปิดใช้หน้าต่าง n ทับบนหน้าต่างอื่นชั่วคราว ซึ่งจะต้องใช้คู่กับ OffTopWindow ก่อนที่จะดำเนินการใดที่เป็นการเปลี่ยนแปลงหน้าต่าง โดยที่ OnTopWindow จะเก็บตำบลที่ของเคอร์เซอร์ของหน้าต่างปัจจุบันไว้ใน -cursx และ -cursy แล้วเปิดหน้าต่าง n ด้วย ActiveWindow

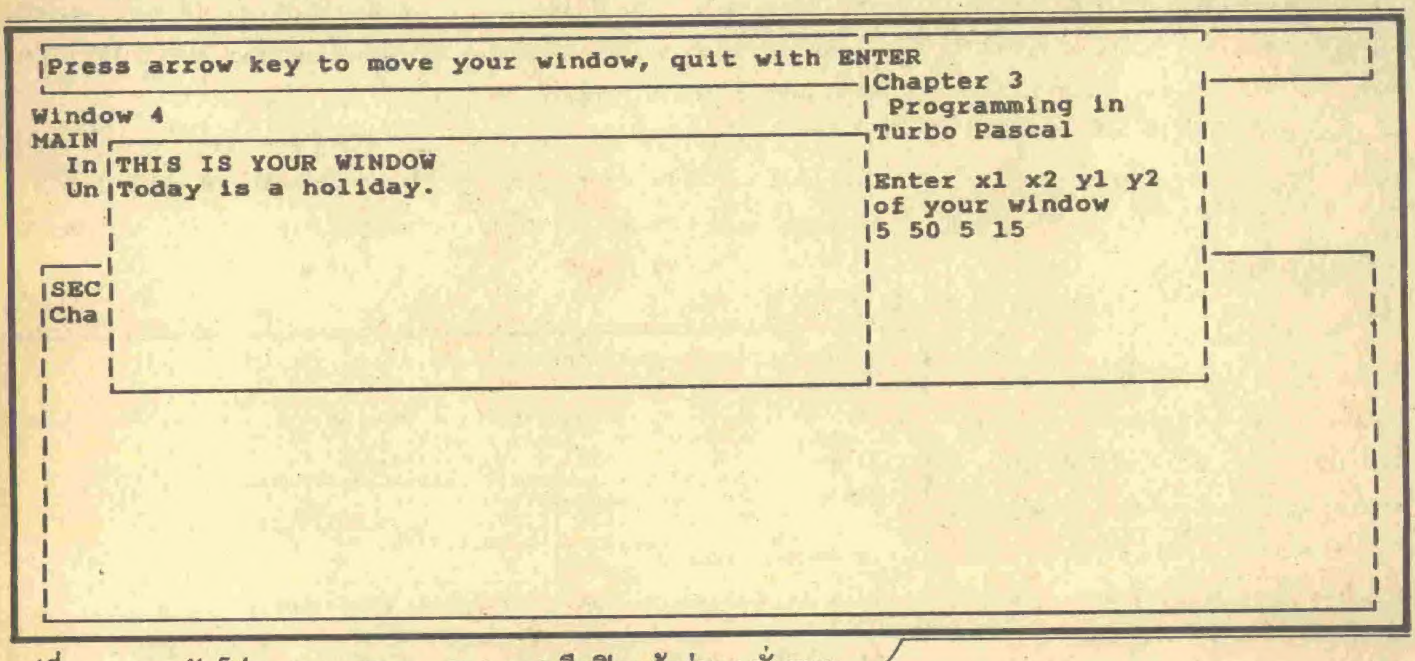
OffTopWindow

ปิดหน้าต่างที่เปิดชั่วคราว แต่ความเป็นจริงเป็นการยกเลิก โดยเปิดหน้าต่างเดิมซึ่งมีหมายเลขหน้าต่างอยู่ที่ชั้นบนสุดของสแต็ก แล้วให้ตำบลที่ของเคอร์เซอร์ตามที่เก็บไว้ใน -cursx และ -cursy

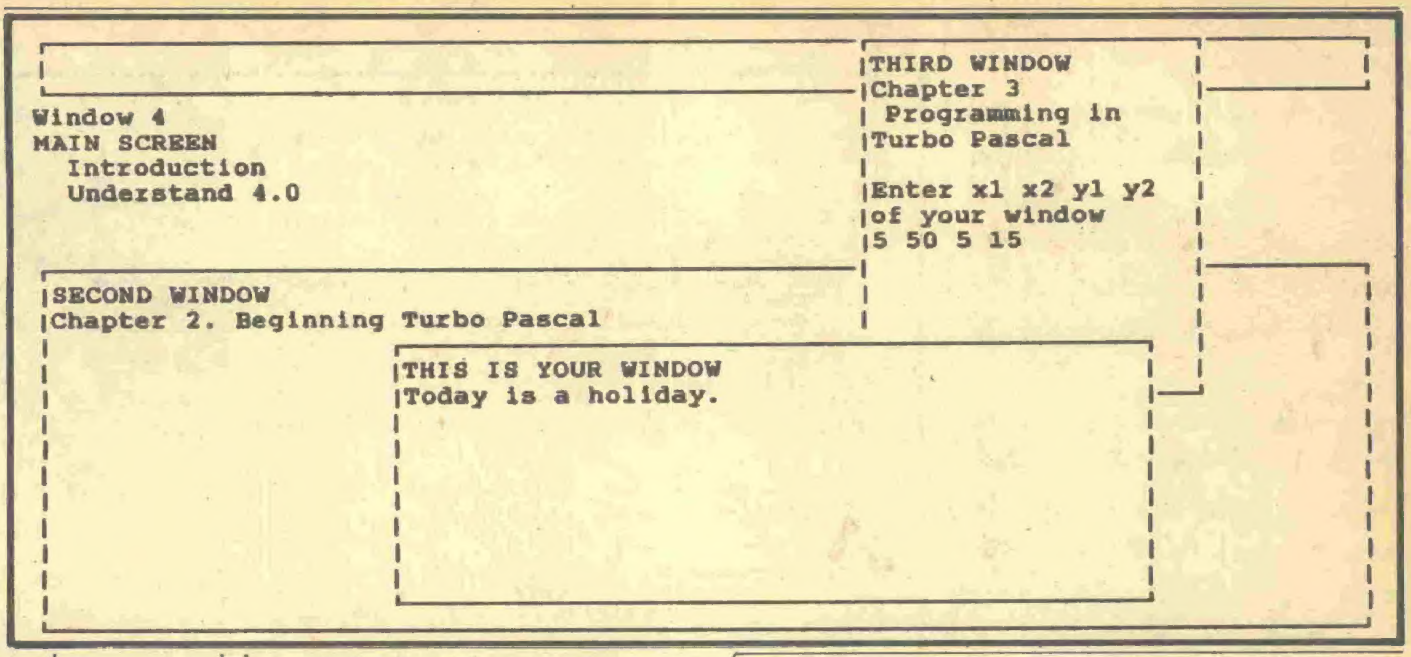
โปรแกรมทั้งสองนี้อยู่ในรูป I6 ซึ่งต้องนำไปป้อนต่อท้าย CloseWindow ในไฟล์ WINDOW.ICL เพราะไฟล์นี้ต่อไปจะทำเป็นยูนิต

M2

```
BEGIN   { MAIN }
  SetUpWindow;

  WriteLn('MAIN SCREEN');
  repeat  readln(astr)  until  astr = '';
  OpenWndIto(3);

  CreateUserWindow;
  OnTopWindow(1);
  Write('Press arrow key to move your window, quit with ENTER');
  OffTopWindow;
  MoveWindowAround;

  CloseNwindow(4);
END.
```



รูปที่ 1 ผลการรันโปรแกรม Demo-Pointer2 ถึงเปิดหน้าต่าง 1 ชั่วคราว



รูปที่ 2 ภายหลังที่เลื่อนหน้าต่างด้วยคีย์ลูกศร ลงล่าง ขวา

MoveWindowAround

อยู่ในรูป D โปรแกรมนี้จะตรวจจับการกดคีย์ ถ้าได้ไบต์แรกเป็นอักขระ #0 คือ null ให้อ่านไบต์ต่อไปจะเป็นรหัสของคีย์พิเศษ แต่เราสนใจเฉพาะคีย์ลูกศร 4 คีย์เท่านั้น โดยให้ค่า X หรือ Y เป็นบวกหรือลบ แล้วเรียกใช้ MoveWindow

MoveWindow (x,y)

ได้เขียนโปรแกรมไว้ใช้ทั่วไปดังในรูป I7 คือจะเลื่อนหน้าต่างบนไปยังจุดที่ต้องการ เช่น ถ้า x = 10 และ y = 5 จะเลื่อนหน้าต่างไปทางขวา 10 อักขระ และลงล่าง 5 บรรทัด การทำงานเริ่มด้วยการให้ค่าจุดโคออร์ดิเนตใหม่แก่ edge ของหน้าต่างบน เสร็จแล้วเก็บตำแหน่งเคอร์เซอร์ แล้วย้ายเทมเพลตของหน้าต่างล่างคือ wnd[old] มาให้แก่เทมเพลตของหน้าต่างบน (actwnd) เท่ากับลบเทมเพลตของหน้าต่างบนเดิม ไม่มีช่องหน้าต่างของตนเอง

จึงต้องลอกเฉพาะช่วงหน้าต่างบนจากจอภาพไปให้ด้วย TranWindow ซึ่งจะลอกไปลงตรงโคออร์ดิเนตใหม่ เท่ากับเป็นการเลื่อนหน้าต่างในเทมเพลตของหน้าต่างบนแล้ว เมื่อนำค่าเทมเพลตนี้กลับมาให้แก่จอภาพใหม่คือ ptrscrn^ := actwnd^ จึงเท่ากับเป็นการเลื่อนหน้าต่างบนจอภาพ

การป้อนโปรแกรมนี้ขอให้ป้อนจาก A2, D และ M2 ส่วน I6 และ I7 ให้ป้อนต่อท้ายในไฟล์ WINDOW.ICL

I6

```
PROCEDURE  OnTopWindow(n : Byte);
BEGIN
  _cursX := WhereX;  _cursY := WhereY;
  ActiveWindow(wnd[n]);
  ClrScr;
END;

PROCEDURE  OffTopWindow;
BEGIN
  ActiveWindow(wnd[stack[sp]]);
  GotoXY( _cursX, _cursY);
END;
```

D

```
PROCEDURE  MoveWindowAround;
CONST   null = #0;  cr = #13;
        up   = #72;  down = #80;  left = #75;  right = #77;
VAR  ch : Char;  x, y : ShortInt;
BEGIN
  REPEAT
    x := 0;  y := 0;
    ch := ReadKey;
    if  ch = null  then  ch := ReadKey;
    case  ch  of
      up     :    y := -1;
      down   :    y :=  1;
      left   :    x := -1;
      right  :    x :=  1;
    end;
    MoveWindow(x,y);
  UNTIL  ch = cr;
END;
```

เลือกหน้าต่าง

โครงงานนี้จะเป็นการสร้างหน้าต่างหลาย ๆ หน้าต่างแล้วให้ผู้ใช้เปิดใช้ทีละหน้าต่าง โดยจะหมุนวนนำหน้าต่างที่

ต้องการมาไว้ด้านบน (ขอให้ดูรูปที่ 3 และที่ 4 ก่อนครับ)

โปรแกรมนี้เริ่มที่รูป A3 ซึ่งคล้ายกับ A1 และ A2 ดังที่ผ่านมา แต่ได้เปลี่ยนค่าบางค่า ส่วนโปรแกรมหลักอยู่ในรูป M3 เริ่มด้วยการเตรียมหน้าต่างตาม MAXWNDW คือ 5 หน้าต่าง แล้วเปิดใช้ทีละหน้าต่างเช่นเคย ดังผลการรันในรูปที่ 3

การเลือกหน้าต่างอยู่ในวงรอบ REPEAT ให้ผู้ใช้เลือกหน้าต่างได้ตั้งแต่หน้าต่าง 1 ถึง หน้าต่าง 5 โดยมอบให้ SelectWindow เป็นผู้ดำเนินการ และจะเป็นการเปิดใช้หน้าต่างนั้นด้วย ซึ่งในโปรแกรมหลักเพียงแต่พิมพ์คำว่า HERE และให้ผู้ใช้ป้อนสตริง 1 สตริง ได้ผลการรันดังรูปที่ 4 จะออกจากวงรอบนี้ได้ด้วยการเลือกหน้าต่าง 0 หากผู้ใช้กดเลขอื่นก็จะไม่ดำเนินการใด รายละเอียดของโปรแกรมย่อยดังจะกล่าวต่อไปนี้คือ

SelectWindow (n)

จะเลือกหน้าต่าง n มาใช้เป็นหน้าต่างบนโดยให้ n อยู่ชั้นบนสุดของสแต็กด้วย อยู่ในรูป I9 เริ่มด้วยการเก็บจอภาพลงแผ่นใสของหน้าต่างที่เป็นชั้นบนสุดขณะนั้น แล้วหมุนวนหมายเลขหน้าต่างในสแต็กให้ n มาอยู่ชั้นบนสุดด้วย CyClingStack หากเรียบร้อยก็จะจัดภาพของหน้าต่างต่าง ๆ ด้วย Rearrange Window ดังแสดงในรูปที่ 4

CyclingStack

ดังที่กล่าวในตอนที่แล้วว่าผมใช้สแต็กเก็บลำดับที่ของจอภาพ ดังนั้นการหมุนวนหน้าต่างจึงเริ่มด้วยการหมุนวนข้อมูล คือหมายเลขหน้าต่างในสแต็กก่อนคือ

1 2 3 4 5

4 5 1 2 3

บรรทัดแรกเป็นหมายเลขหน้าต่าง

I7

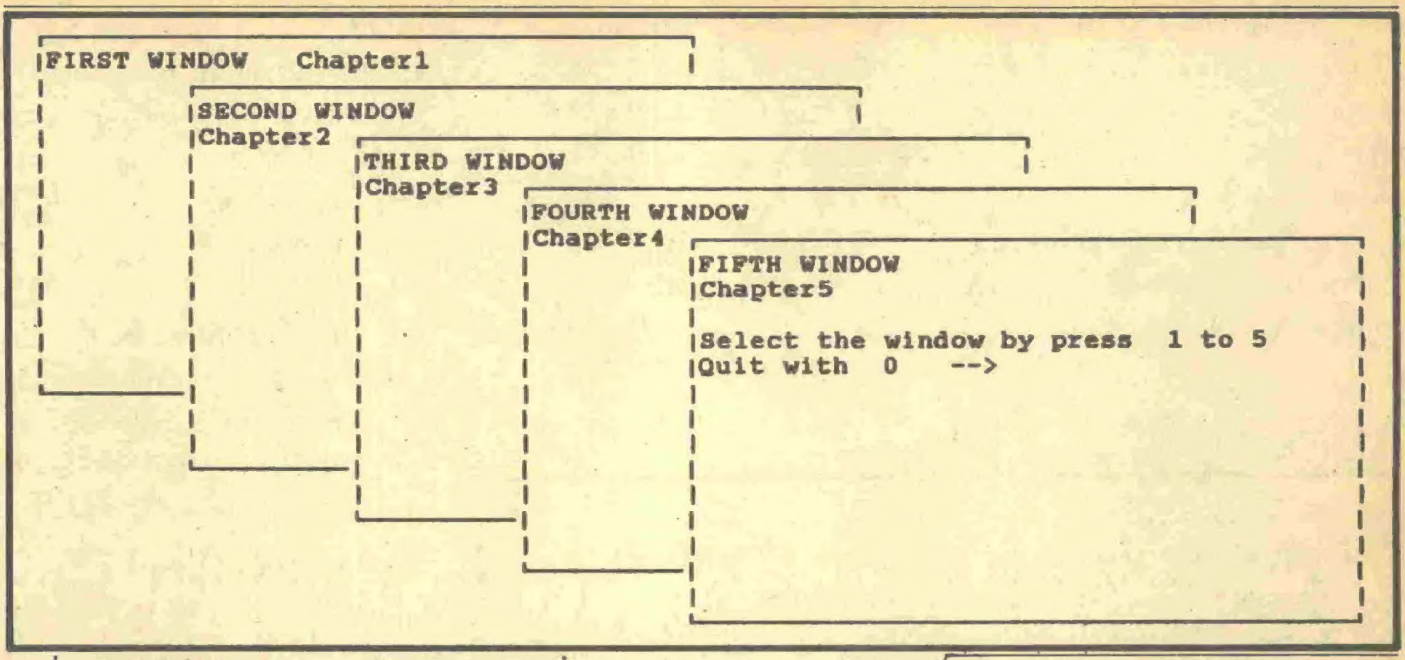
```
(*....................... Move Window .....................*)

PROCEDURE  TranWindow(VAR srce, dstn : WndPtr);
VAR    rows,cols,rowd,cold : Byte;
BEGIN
  rowd := dstn^.edge[3];
  FOR  rows := srce^.edge[3] TO srce^.edge[4]  DO
    BEGIN
    cold := dstn^.edge[1];
    FOR  cols := srce^.edge[1] TO srce^.edge[2]  DO
      BEGIN
      dstn^.scr[rowd,cold] := srce^.scr[rows,cols];
      cold := cold + 1;
      END;
    rowd := rowd + 1;
    END;
  dstn^.curX := srce^.curX;  dstn^.curY := srce^.curY;
END;

PROCEDURE  MoveWindow(x,y : ShortInt);
VAR    old : Byte;
BEGIN
  WITH  actwnd^  DO
    BEGIN  edge[1] := edge[1] + x;  edge[2] := edge[2] + x;
           edge[3] := edge[3] + y;  edge[4] := edge[4] + y;
           curX := WhereX;    curY := WhereY;                END;
  old := stack[sp-1];
  actwnd^.scr := wnd[old]^.scr;
  TranWindow(ptrscrn, actwnd);
  ptrscrn^ := actwnd^;
  ActiveWindow(actwnd);
END;
```

A3

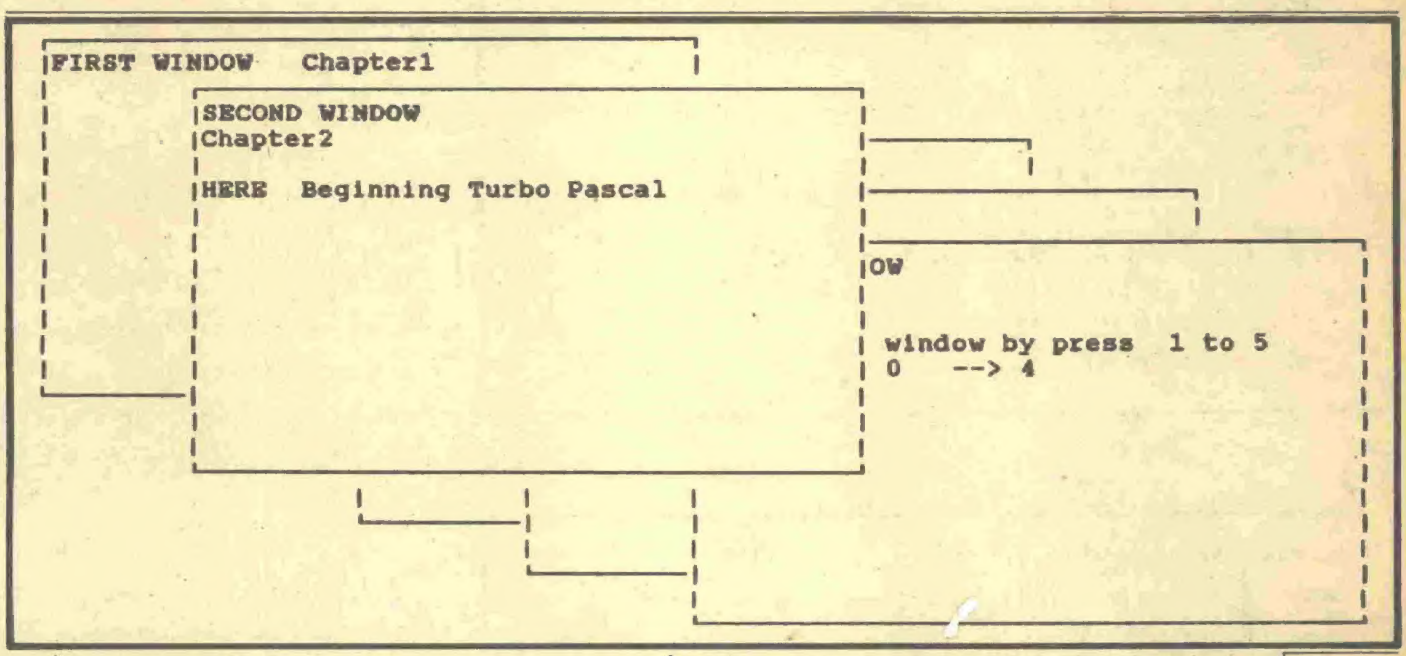
```
PROGRAM  DemoPointer3(input,output);
USES     crt;

(*-----------------  CONSTANT  for WINDOW.ICL  -----------*)
CONST    MAXWNDW = 5;  MAXSTCK = 10;
TYPE     edgeT   = ARRAY[1..4] OF Byte;
         WindowSizeT = ARRAY[0..MAXWNDW] OF edgeT;
CONST    wndSize : WindowSizeT =
          ((1,80,1,25), (1,40,1,15), (10,50,3,18),
           (20,60,5,20), (30,70,7,22), (40,80,9,24) );
{$I WINDOW.ICL}
(*-------------------------------------------------------*)
TYPE     promptT  = ARRAY[1..MAXWNDW] OF String[14];
CONST    promp    : promptT =
          ('FIRST WINDOW ','SECOND WINDOW','THIRD WINDOW',
           'FOURTH WINDOW','FIFTH WINDOW');
{$I WINDOW2.ICL}
(*-------------------------------------------------------*)
```

M3

```
var   num : Byte;
BEGIN   { MAIN }
  SetUpWindow;

  WriteLn('MAIN SCREEN');
  repeat  readln(astr)  until  astr = '';
  OpenWndlto(5);

  REPEAT
    WriteLn('Select the window by press  1 to 5');
    Write('Quit with  0   --> ');
    ReadLn(num);
    IF  num  IN [1..MAXWNDW]  THEN
      BEGIN
      SelectWindow(num);
      Write('HERE  ');
      ReadLn(astr);
      END
    ELSE  WriteLn('Sorry wrong number');
  UNTIL  num = 0;
END.
```



รูปที่ 3 ผลการรันโปรแกรม Demo-Pointer3 เมื่อสร้างหน้าต่างครบ 5 หน้าต่าง



รูปที่ 4 ผลการรันโปรแกรม Demo-Pointer3 ภายหลังที่ผู้ใช้เลือกหน้าต่าง 2 และพิมพ์ข้อความว่า Beginning Turbo Pascal

ในสแต็กตามที่เราเปิดใช้ สมมติว่าผู้ใช้เลือกหน้าต่าง 3 ผมจะต้องหมุนวนข้อมูลในสแต็กให้เป็นดังบรรทัดที่สอง

ขั้นตอนวิธีหมุนวนข้อมูล

การหมุนวนข้อมูลเพียงไม่กี่ค่าในสแต็กดังที่กล่าวมานี้ไม่มีปัญหาใด จะดำเนินการอย่างไรก็ได้ เช่น หมุนทีละตัวจนถึงค่าที่ต้องการ เช่น

1 2 3 4 5

5 1 2 3 4

4 5 1 2 3

ซึ่งการดำเนินการดังกล่าวนี้จะต้องเคลื่อนย้ายข้อมูลเป็นจำนวนครั้งเท่ากับ (จำนวนข้อมูลทั้งหมด + 1) × ลำดับที่ของข้อมูลที่ต้องการ จากตัวอย่างที่ยกมานี้จะต้องเคลื่อนย้ายข้อมูลเท่ากับ (5+1) ×2 = 12 ครั้ง ดูไม่มากเท่าไร แต่ถ้ามีข้อมูลเป็น 100 จำนวนครั้งที่เคลื่อนย้ายข้อมูลน้อยที่สุดคือ 101×1 = 101 ครั้ง มากที่สุดคือ 101×99 = 9999 ครั้ง

จึงมีผู้กำหนดขั้นตอนวิธี (algoithm) ไว้ให้ดำเนินการและใช้กันมากในโปรแกรมประเภทเวิร์ดโปรเซสเซอร์ ผมจึง

ถือโอกาสนำมาเสนอ การเลื่อนข้อมูลแบบนี้จะมีจำนวนครั้งในการเคลื่อนย้ายข้อมูลประมาณ 3 เท่าของจำนวนข้อมูลทั้งหมด โดยไม่เกี่ยวกับตำแหน่งที่ของข้อมูลที่ต้องการ

การอธิบายขั้นตอนวิธีนี้จะต้องอธิบายประกอบตัวอย่าง โดยสมมติว่ามีข้อมูลอยู่ 9 ตัวคือ 1 ถึง 9 และต้องการหมุนวนให้เลข 4 ไปแทนที่เลข 9 จะดำเนินการได้ดังนี้

1 2 3 4 5 6 7 8 9

4 3 2 1 9 8 7 6 5

5 6 7 8 9 1 2 3 4

คือสลับที่ 1 กับ 4, 2 กับ 3 พวกหนึ่ง และ 5 กับ 9, 6 กับ 8 อีกพวกหนึ่ง ได้ผลดังบรรทัดที่สอง แล้วสลับที่รวมคือ 4 กับ 5, 3 กับ 6, 2 กับ 7 และ 1 กับ 8 จะได้ผลดังบรรทัดที่สาม จากตัวอย่างที่ยกมานี้จะเห็นว่าผมสลับที่ข้อมูลทั้งหมด 8 ครั้ง แต่ในการสลับที่ข้อมูล 1 ครั้งจะต้องเคลื่อนย้ายข้อมูล 3 ครั้ง จึงกล่าวว่าจะเคลื่อนย้ายข้อมูลประมาณ 3 เท่าของข้อมูลทั้งหมด

นำไปเขียนเป็นโพรซีเดอร์ CyClingStack รูป I8 คือในวงรอบ REPEAT จะค้นหาหมายเลขของหน้าต่างที่ผ่านค่ามาให้ dat ว่าอยู่ตรงไหนของสแต็กแล้วนำไปสลับที่กันดังกล่าวมาแล้ว

I8

```
(*.....,.....  Select Window  .....*)

PROCEDURE  CyclingStack(dat : Byte; VAR OK : Boolean);
VAR   cntr, hi, lo, tem : Byte;
  PROCEDURE  Swap(var lo,hi : Byte);
  BEGIN
    tem := stack[lo];
    stack[lo] := stack[hi];
    stack[hi] := tem;
    lo := lo + 1;  hi := hi - 1;
  END;

BEGIN
  OK := TRUE;  cntr := 0;
  REPEAT  cntr := cntr + 1;
  UNTIL   (dat = stack[cntr]) OR  (cntr > sp);
  IF  cntr < sp  THEN
    BEGIN
    lo := 1; hi := cntr;
    if lo < hi then repeat Swap(lo,hi) until lo >= hi;

    lo := cntr + 1; hi := sp;
    if lo < hi then repeat Swap(lo,hi) until lo >= hi;

    lo := 1; hi := sp;
    if lo < hi then repeat Swap(lo,hi) until lo >= hi;
    END
  ELSE  OK := FALSE;
END;
```

I9

```
PROCEDURE  RearrangeWindow;
VAR  I : Byte;
BEGIN
  New(temw);
  temw^ := wnd[0]^ ;
  FOR  I := 1 TO sp  DO
    BEGIN
    CopyWindow(wnd[stack[I]], temw);
    wnd[stack[I]]^ := temw^ ;
    END;
  actwnd := wnd[stack[sp]];
  ptrscrn^ := actwnd^ ;
  ActiveWindow(actwnd);
  Dispose(temw);
END;

PROCEDURE  SelectWindow(n : Byte);
VAR  OK : Boolean;
BEGIN
  SaveScreen(actwnd);
  CyClingStack(n, OK);
  IF  OK  THEN  RearrangeWindow;
END;

(*.....*)
```

RearrangeWindow

เมื่อหมุนวนหมายเลขหน้าต่างในสแต็กแล้ว ขั้นตอนต่อไปคือจัดลำดับภาพในแผ่นใสของแต่ละหน้าต่างให้สอดคล้องกัน เริ่มต้นด้วยการกำหนดหน้าต่างชั่วคราวคือ temw แล้วย้ายภาพจากแผ่นใสของจอภาพมาให้

จากนั้นก็จะดำเนินการลอกภาพเฉพาะตรงช่องหน้าต่างจากแผ่นใสของหน้าต่างล่างสุดให้แก่ temw เสร็จแล้วถ่ายภาพกลับไปให้แผ่นใสของหน้าต่างนั้นอีกที ทำซ้ำเช่นนี้กับหน้าต่างชั้นบนขึ้นมา จะได้ภาพซ้อนกันจนถึงหน้าต่างชั้นบนสุด เสร็จแล้วจึงให้ภาพจากแผ่นใสของหน้าต่างชั้นบนสุดแก่จอภาพ แล้วเปิดใช้หน้าต่าง

ในโพรซีเดอร์นี้ได้แสดงการยกเลิกข้อมูลในฮิพด้วยคำสั่ง Dispose ซึ่งความจริงสำหรับกรณีนี้ไม่จำเป็นจะต้องยกเลิก แต่ต้องไม่ New (temw) ในโพรซีเดอร์นี้

การป้อนโปรแกรมนี้ให้ป้อน A3 กับ M3 ส่วน I8 และ I9 ให้นำไปป้อนต่อท้ายในไฟล์ WINDOW.ICL

ผมคิดว่าคุณผู้อ่านคงสนุกกับจอภาพพอสมควร หากมีโอกาสผมจะนำมาเสนอต่อ เพราะตอนนี้ต้องการเน้นที่พอยน์เตอร์ ซึ่งยังเหลืออีกเรื่องหนึ่งคือพอยน์เตอร์กับข้อมูลแบบไดนามิกส์ก็คงต้องขอยกไปฉบับหน้าครับ